

AI Assisted Coding

Assignment 6.5

Y. Sahasra

2303A51269

Batch 19

Task 1: AI-Based Code Completion for Conditional Eligibility Check Prompt:

Generate Python code to check voting eligibility based on age and citizenship.

Code & Output:

The screenshot shows a code editor interface with two panes. The left pane displays a Python script named 'Assignment_6.5.py' with the following code:

```
1  #Task 1: AI-Based Code Completion for Conditional Eligibility Check
2  # Generate Python code to check voting eligibility based on age and
3  # citizenship.
4  age = int(input("Enter your age: "))
5  citizenship = input("Are you a citizen? (yes/no): ").lower()
6
7  if age >= 18 and citizenship == "yes":
8      print("You are eligible to vote.")
9  else:
10     print("You are not eligible to vote.")
```

The right pane shows the terminal output of running the script:

```
Are you a citizen? (yes/no): yes
Are you a citizen? (yes/no): yes
You are not eligible to vote.
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5.py"
Enter your age: 18
Are you a citizen? (yes/no): no
You are not eligible to vote.
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5.py"
Enter your age: 18
Are you a citizen? (yes/no): yes
You are eligible to vote.
PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

Explanation:

The AI-generated code uses conditional statements to check voting eligibility. It verifies whether the age is 18 or above and whether the user is a citizen. Both conditions must be true for eligibility. This demonstrates correct use of conditional logic generated through AI-based code completion.

Task 2: AI-Based Code Completion for Loop-Based String Processing

Prompt:

Generate Python code to count vowels and consonants in a string using a loop.

Code & Output:

The screenshot shows a code editor interface with two main panes. The left pane displays a Python script named `Assignment_6.5.py`. The code is a solution to a task involving vowel and consonant counting. It uses a loop to iterate through each character of the input string, checks if it's alphabetic using `char.isalpha()`, and then checks if it's a vowel from the set `vowels = "aeiouAEIOU"`. If it is a vowel, the `vowel_count` variable is incremented; otherwise, the `consonant_count` variable is incremented. Finally, the script prints the counts. The right pane shows a terminal window titled "AI Assisted Coding" with the command `python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5\Assignment_6.5.py"` running. The terminal output shows the user entering "Hello World" and the script outputting "Vowels: 3, Consonants: 7".

```
Assignment_6.5.py
Assignment_6.5 > Assignment_6.5.py > ...
11  # Task 2: AI-Based Code Completion For Loop-Based String Processing"
12  # Generate python code to count vowels and consonants in a string using a loo
13  input_string = input("Enter a string: ")
14  vowel_count = 0
15  consonant_count = 0
16  vowels = "aeiouAEIOU"
17  for char in input_string:
18      if char.isalpha():
19          if char in vowels:
20              vowel_count += 1
21          else:
22              consonant_count += 1
23  print(f"Vowels: {vowel_count}, Consonants: {consonant_count}")
24

PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5\Assignment_6.5.py"
Enter a string: Hello World
Vowels: 3, Consonants: 7
PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

Explanation:

The AI-generated code processes the input string using a loop. Each character is checked to determine whether it is a vowel or a consonant. Alphabetic characters are counted correctly, while non-letter characters are ignored. The output verifies that the logic works as expected.

Task 3: AI-Assisted Code Completion Reflection Task Prompt:

Generate a Python program for a library management system using classes, loops, and conditional statements.

Code & Output:

```
Assignment_6.5.py
Assignment_6.5 > Assignment_6.5.py > ...
24
25 # Task 3: AI-Assisted Code Completion Reflection Task
26 # Generate a Python program for a library management system using classes,
27 # loops, and conditional statements.
28
29 class Book:
30     def __init__(self, title, author):
31         self.title = title
32         self.author = author
33         self.is_borrowed = False
34
35     def borrow(self):
36         if not self.is_borrowed:
37             self.is_borrowed = True
38             return True
39         return False
40
41     def return_book(self):
42         if self.is_borrowed:
43             self.is_borrowed = False
44         return True
45     return False
46
47 class Library:
48     def __init__(self):
49         self.books = []
50
51     def add_book(self, book):
52         self.books.append(book)
53
54     def display_books(self):
55         for idx, book in enumerate(self.books):
56             status = "borrowed" if book.is_borrowed else "Available"
57             print(f"[{idx + 1}]. {book.title} by {book.author} - {status}")
58
59     def borrow_book(self, index):
60         if 0 <= index < len(self.books):
61             if self.books[index].borrow():
62                 print(f"You have borrowed '{self.books[index].title}'.")
63
64             else:
65                 print(f"{self.books[index].title} is already borrowed.")
66
67             else:
68                 print("Invalid book index.")
69
70     def return_book(self, index):
71         if 0 <= index < len(self.books):
72             if self.books[index].return_book():
73                 print(f"You have returned '{self.books[index].title}'.")
74             else:
75                 print(f"{self.books[index].title} was not borrowed.")
76             else:
77                 print("Invalid book index.")
78
79     def main():
80         library = Library()
81         library.add_book(Book("1984", "George Orwell"))
82         library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
83         library.add_book(Book("The Great Gatsby", "F. Scott Fitzgerald"))
84
85         while True:
86             print("\nLibrary Menu:")
87             print("1. Display Books")
88             print("2. Borrow Book")
89             print("3. Return Book")
90             print("4. Exit")
91             choice = input("Enter your choice: ")
92
93             if choice == '1':
94                 library.display_books()
95             elif choice == '2':
96                 index = int(input("Enter the book index to borrow: ")) - 1
97                 library.borrow_book(index)
98             elif choice == '3':
99                 index = int(input("Enter the book index to return: ")) - 1
100                library.return_book(index)
```

PS E:\3rd Year\2nd Sem\AI Assisted coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5\Assignment_6.5.py"

1. Display Books
2. Borrow Book
3. Return Book
4. Exit

Enter your choice: 2
Enter the book index to borrow: 3
You have borrowed 'The Great Gatsby'.

Library Menu:
1. Display Books
2. Borrow Book
3. Return Book
4. Exit

Enter your choice: 1
1. 1984 by George Orwell - Available
2. To Kill a Mockingbird by Harper Lee - Available
3. The Great Gatsby by F. Scott Fitzgerald - Borrowed

Library Menu:
1. Display Books
2. Borrow Book
3. Return Book
4. Exit

Enter your choice: 3
Enter the book index to return: 3
You have returned 'The Great Gatsby'.

Library Menu:
1. Display Books
2. Borrow Book
3. Return Book
4. Exit

Enter your choice: 1
1. 1984 by George Orwell - Available
2. To Kill a Mockingbird by Harper Lee - Available
3. 1984 by George Orwell - Available
4. To Kill a Mockingbird by Harper Lee - Available
5. The Great Gatsby by F. Scott Fitzgerald - Available

```
Assignment_6.5.py
Assignment_6.5 > Assignment_6.5.py > ...
44
45 class Library:
46     def borrow_book(self, index):
47         if 0 <= index < len(self.books):
48             if self.books[index].borrow():
49                 print(f"{self.books[index].title} is already borrowed.")
50             else:
51                 print("Invalid book index.")
52
53     def return_book(self, index):
54         if 0 <= index < len(self.books):
55             if self.books[index].return_book():
56                 print(f"You have returned '{self.books[index].title}'.")
57             else:
58                 print(f"{self.books[index].title} was not borrowed.")
59
60     def main():
61         library = Library()
62         library.add_book(Book("1984", "George Orwell"))
63         library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
64         library.add_book(Book("The Great Gatsby", "F. Scott Fitzgerald"))
65
66         while True:
67             print("\nLibrary Menu:")
68             print("1. Display Books")
69             print("2. Borrow Book")
70             print("3. Return Book")
71             print("4. Exit")
72             choice = input("Enter your choice: ")
73
74             if choice == '1':
75                 library.display_books()
76             elif choice == '2':
77                 index = int(input("Enter the book index to borrow: ")) - 1
78                 library.borrow_book(index)
79             elif choice == '3':
80                 index = int(input("Enter the book index to return: ")) - 1
81                 library.return_book(index)
```

PS E:\3rd Year\2nd Sem\AI Assisted coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5\Assignment_6.5.py"

Enter the book index to borrow: 3
You have borrowed 'The Great Gatsby'.

Library Menu:
1. Display Books
2. Borrow Book
3. Return Book
4. Exit

Enter your choice: 1
1. 1984 by George Orwell - Available
2. To Kill a Mockingbird by Harper Lee - Available
3. 1984 by George Orwell - Available
4. To Kill a Mockingbird by Harper Lee - Available
5. The Great Gatsby by F. Scott Fitzgerald - Available

Library Menu:
1. Display Books
2. Borrow Book
3. Return Book
4. Exit

Enter your choice: 3
Enter the book index to return: 3
You have returned 'The Great Gatsby'.

Library Menu:
1. Display Books
2. Borrow Book
3. Return Book
4. Exit

Enter your choice: 1
1. 1984 by George Orwell - Available
2. To Kill a Mockingbird by Harper Lee - Available
3. 1984 by George Orwell - Available
4. To Kill a Mockingbird by Harper Lee - Available
5. The Great Gatsby by F. Scott Fitzgerald - Available

The screenshot shows a code editor with two tabs: 'Assignment_6.5.py' and 'Code'. The 'Assignment_6.5.py' tab contains Python code for a library management system. The 'Code' tab shows the terminal output of running the program.

```
Assignment_6.5.py
Assignment_6.5 > Assignment_6.5.py > main
73     def main():
74         library = Library()
75         library.add_book(Book("1984", "George Orwell"))
76         library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
77         library.add_book(Book("The Great Gatsby", "F. Scott Fitzgerald"))
78
79         while True:
80             print("Library Menu:")
81             print("1. Display Books")
82             print("2. Borrow Book")
83             print("3. Return Book")
84             print("4. Exit")
85             choice = input("Enter your choice: ")
86
87             if choice == '1':
88                 library.display_books()
89             elif choice == '2':
90                 index = int(input("Enter the book index to borrow: ")) - 1
91                 library.borrow_book(index)
92             elif choice == '3':
93                 index = int(input("Enter the book index to return: ")) - 1
94                 library.return_book(index)
95             elif choice == '4':
96                 print("Exiting the library system.")
97                 break
98             else:
99                 print("Invalid choice. Please try again.")
100
101     if __name__ == "__main__":
102         main()

PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5\Assignment_6.5.py"
2. Borrow Book
3. Return Book
4. Exit
Enter your choice: 3
Enter the book index to return: 3
You have returned "The Great Gatsby".

Library Menu:
1. Display Books
2. Borrow Book
3. Return Book
4. Exit
Enter your choice: 1
1. 1984 by George Orwell - Available
2. To Kill a Mockingbird by Harper Lee - Available
1. 1984 by George Orwell - Available
2. To Kill a Mockingbird by Harper Lee - Available
3. The Great Gatsby by F. Scott Fitzgerald - Available

Library Menu:
1. Display Books
2. Borrow Book
3. Return Book
4. Exit
Enter your choice: 4
Exiting the library system.
PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

Explanation:

The AI-generated program uses a class to represent a library and includes loops and conditional statements for menu-driven interaction. The loop allows continuous user input, and conditionals control program flow. The program correctly demonstrates AI-assisted use of object-oriented programming concepts.

Reflection on AI-Assisted Coding:

The AI tool generated a complete and functional program quickly. While the logic is correct, the code can be further improved with input validation and advanced features. This task shows that AI is useful for speeding up development but still requires human review and optimization.

Task 4: AI-Assisted Code Completion for Class-Based Attendance System

Prompt:

Generate a Python class to mark and display student attendance using loops.

Code & Output:

```

Assignment_6.5.py > Assignment_6.5.py > ...
102
103     "Task 4: AI-Assisted Code completion for Class-Based Attendance System"
104     # Generate a Python class to mark and display student attendance using loops.
105     class AttendanceSystem:
106         def __init__(self):
107             self.attendance = {}
108
109         def mark_attendance(self, student_name):
110             self.attendance[student_name] = "Present"
111
112         def display_attendance(self):
113             print("Attendance Record:")
114             for student, status in self.attendance.items():
115                 print(f"{student}: {status}")
116
117     def main():
118         attendance_system = AttendanceSystem()
119         while True:
120             name = input("Enter student name to mark attendance (or 'exit' to finish): ")
121             if name.lower() == 'exit':
122                 break
123             attendance_system.mark_attendance(name)
124         attendance_system.display_attendance()
125     if __name__ == "__main__":
126         main()

```

Explanation:

The AI-generated attendance system uses a class to store attendance data. A loop is used to take multiple student entries, and another loop displays the attendance records. The code works correctly and demonstrates class-based AI code completion.

Task 5: AI-Based Code Completion for Conditional Menu Navigation Prompt:

Generate a Python program using loops and conditionals to simulate an ATM menu.

Code & Output:

```

Assignment_6.5.py > Assignment_6.5.py > ...
125
126     "Task 5: AI-Based Code Completion for Conditional Menu Navigation"
127     # Generate a Python program using loops and conditionals to simulate an ATM
128     # menu.
129     balance = 1000.0
130     while True:
131         print("\nATM Menu:")
132         print("1. Check Balance")
133         print("2. Deposit Money")
134         print("3. Withdraw Money")
135         print("4. Exit")
136         choice = input("Enter your choice: ")
137
138         if choice == '1':
139             print(f"Your current balance is: ${balance:.2f}")
140         elif choice == '2':
141             amount = float(input("Enter amount to deposit: "))
142             if amount > 0:
143                 balance += amount
144                 print(f"${amount:.2f} deposited successfully.")
145                 print(f"Your current balance is: ${balance:.2f}")
146             else:
147                 print("Invalid amount. Please enter a positive value.")
148         elif choice == '3':
149             amount = float(input("Enter amount to withdraw: "))
150             if 0 < amount <= balance:
151                 balance -= amount
152                 print(f"${amount:.2f} withdrawn successfully.")
153                 print(f"Your current balance is: ${balance:.2f}")
154             else:
155                 print("Invalid amount or insufficient balance.")
156         elif choice == '4':
157             print("Exiting the ATM. Thank you!")
158             break
159         else:
160             print("Invalid choice. Please try again.")

```

Explanation:

The AI-generated ATM program uses a loop to display the menu repeatedly and conditional statements to handle user choices. The logic correctly updates the balance and prevents invalid withdrawals. This task demonstrates effective AI-based code completion for menudriven programs.

Final Conclusion:

This experiment shows how AI-based code completion tools can generate useful Python code involving classes, loops, and conditionals. While AI speeds up development, developers must still review logic, handle edge cases, and ensure ethical and responsible use of Algenerated code.